

CSR1000v HA冗余在Microsoft Azure的部署指南与AzureCLI 2.0

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[目标](#)

[拓扑](#)

[Network Diagram](#)

[术语](#)

[限制](#)

[配置](#)

[概述](#)

[步骤1.安装AzureCLI 2.0。](#)

[步骤2.创建资源组。](#)

[步骤3.创建Vnet。](#)

[步骤4.创建路由表。](#)

[步骤5.创建子网。](#)

[步骤6.创建一个CSR1000v路由器。](#)

[步骤7.创建第二个CSR1000v路由器。](#)

[步骤8.用在第6.步的同一个程序创建主机VM。此示例使用UbuntuLTS。](#)

[步骤9.添加路由到路由表和VMs。](#)

[步骤10.配置CSR1000v路由器。](#)

[验证高可用性](#)

[Troubleshoot](#)

[Related Information](#)

Introduction

本文提供一个逐步配置指南关于怎样配置高可用性的CSR1000v路由器在与AzureCLI 2.0的Microsoft天蓝色的网云。它打算产生用户HA实用的知识和能力配置一张功能完备的试验床。

有配置在Azure的镜像的多种方法，并且多数用户的最熟悉的方法是通过Web门户。然而，一旦熟悉它，AzureCLI是一个快速和强大的工具。

关于关于Azure的更加详细的背景，如何通过Web门户配置CSR1000v和HA，请参见[Microsoft Azure](#)和相关信息部分[Cisco CSR 1000v部署指南](#)。

Prerequisites

Requirements

Cisco 建议您了解以下主题：

- Microsoft Azure帐户
- 2个CSR1000v和1台Windows/Linux虚拟机
- AzureCLI 2.0

Components Used

本文的信息根据Cisco IOS-XE® Denali 16.7.1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. 如果您的网络实际，请保证您了解所有命令的潜在影响。

目标

配置2 CSR1000v路由器和1个VM (窗口/Linux)。模拟从专用的datacenter (VM)的持续数据流到互联网(8.8.8.8)。模拟HA故障切换并且注意到HA通过确认成功天蓝色的路由表有从CSR-A的交换数据流到CSR B的专用接口。

拓扑

为了充分了解拓扑和设计是重要的在配置开始前。这帮助稍后排除所有潜在问题故障。

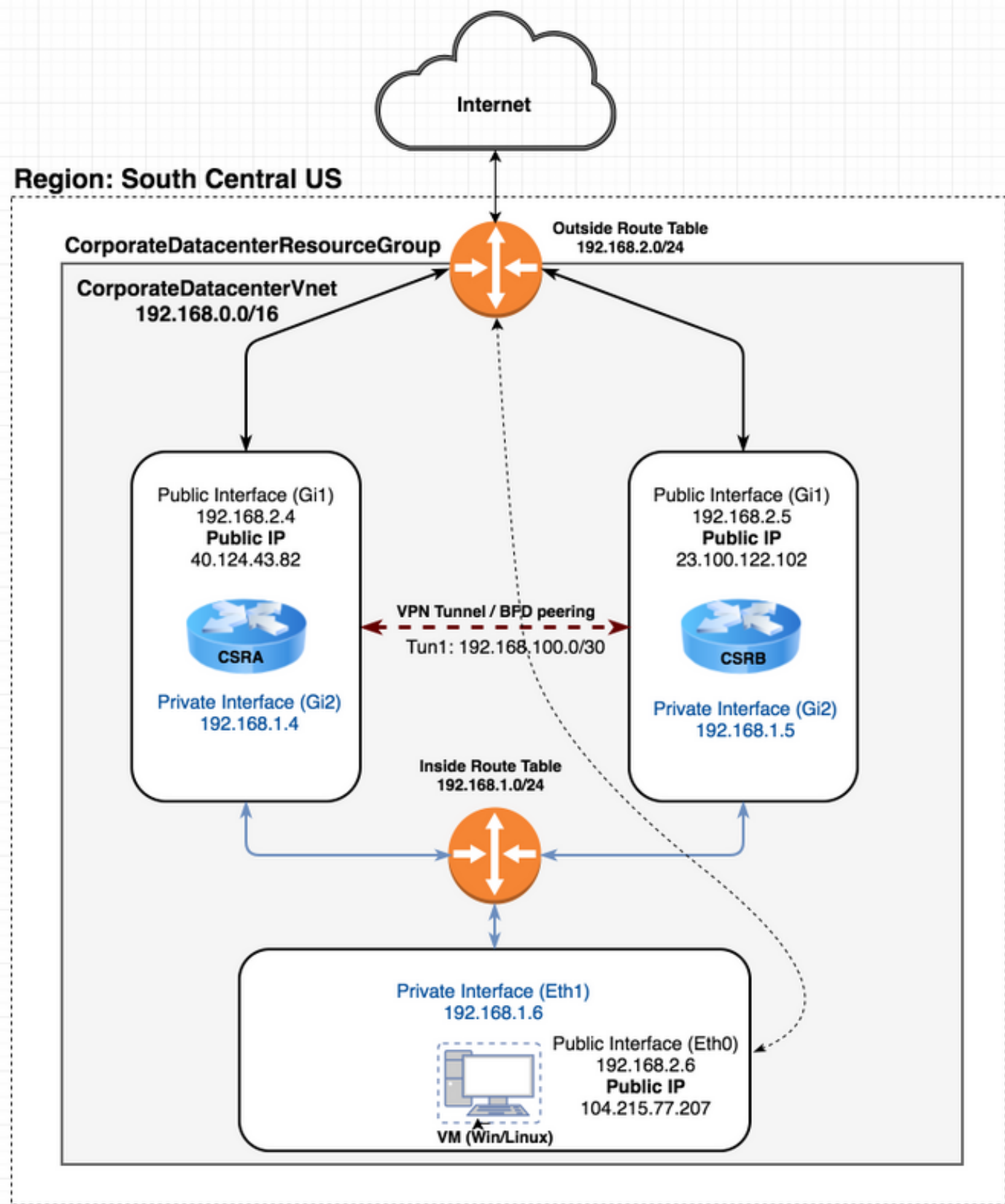
可以有HA配置多种方案根据用户需求的。对于此示例，请用这些设置配置HA冗余：

- 1x -地区(中南部的美国)
- 1x -资源组(CorporateDatacenterResourceGroup)
- 1x - Vnet (CorporateDatacenterVnet)
- 6x -网络接口(3x内部面对和3x外部面对)
- 2x -路由表(InsideRoutetable和OutsideRoutetable)
- 2x - CSR1000v路由器(Cisco IOS XE ® Denali 16.7.1)
- 1x - VM (Linux/Windows)

此刻，互联网访问通过公共接口在VM留给启用，以便您能访问和配置它。通常，所有正常数据流应该流经专用的路由表。在VM的公共接口可以以后被禁用，以便数据流没有偶然地漏。

数据流模拟通过连接进行从VM的专用接口→在路由表→ CSRA → 8.8.8.8里面。在故障切换方案中，请注意专用的路由表交换路由指向CSR B的专用接口。

Network Diagram



术语

- 资源组-这是Azure的一个方式能记录所有您的资源类似虚拟机和vnets。这通常用于管理所有项目和记录充电。
- Vnet -一个虚拟网络。(类似于在AW术语的VPC)
- 路由表-这包含子网的规则，并且能转发特定的流量到IP地址或操作类似VPN终端。

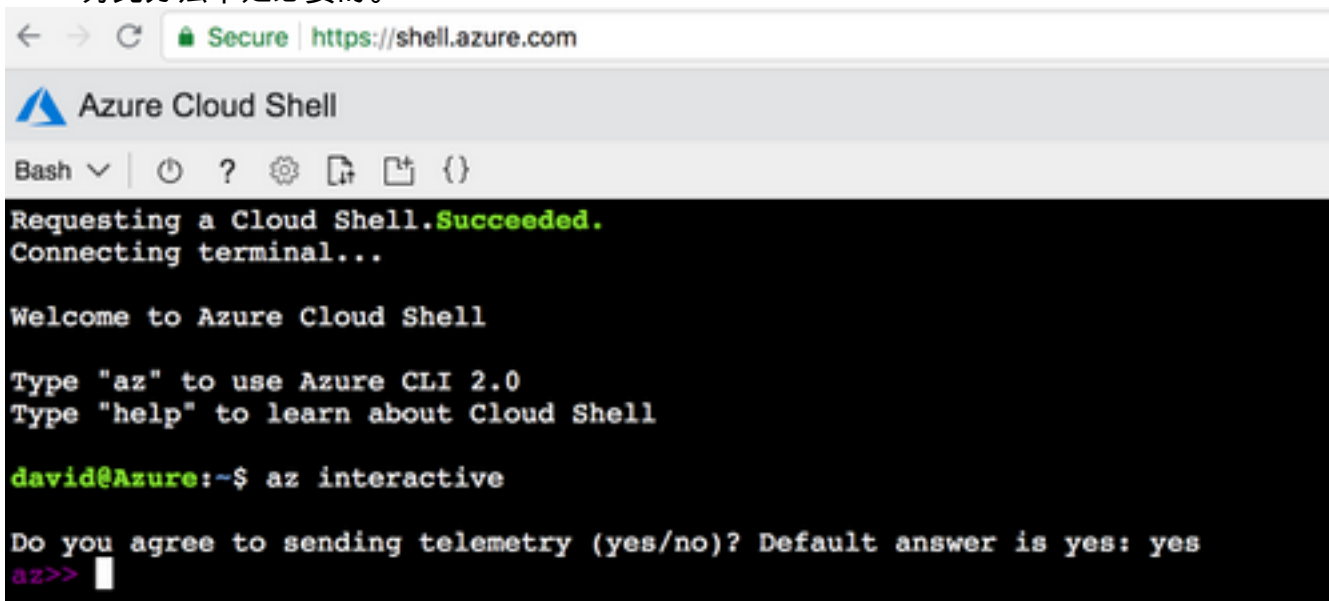
限制

- Azure可能介绍大致在HA故障切换的—40-50秒的延迟。

配置

有配置在Azure的VM的一些个方法：

1. [Web门户](#)-在cisco.com的HA文档
2. Powershell -管理天蓝色的资源的Line命令基本模型。
3. [AzureCLI 2.0](#) -并且基于的line命令。它是开放源和写在Python并且需要在您的本地系统上安装。为了写作本文， AzureCLI 2.0是新版本。
4. [天蓝色的Cloud Shell](#) -选择**打击shell**选项而不是**Powershell**选项通过shell使用AzureCLI。安装为此方法不是必要的。



Powershell和AzureCLI是类似的，但是AzureCLI的命令是更加直接的。两个在Windows能运行，MacOS，Linux。参考[选择Azure和并行的Azure CLI和PowerShell命令的正确的凿出的装饰](#)比较的。

对于此示例，请配置所有资源用AzureCLI或Cloud Shell。 AzureCLI可以在MacOS、Windows或者Linux上安装与有些不同的步骤。 没有在配置上的区别通过程序的其余在AzureCLI和Azure Cloud Shell之间的。

```
redundancy
cloud provider azure 100
bfd peer
route-table
default-gateway ip
cidr ip
app-key
subscription-id
app-id
tenant-id
resource-group
```

Note:此模板是有用记录以后用于配置在CSR的HA的所有ID和设置。

概述

步骤1.安装AzureCLI 2.0。

1. 遵从Windows、MacOS或者Linux的安装步骤在[AzureCLI 2.0](#)文档。

2. MacOS :

```
$ brew update && brew install azure-cli
```

3. 登陆对Azure并且遵从指令验证您的会话。

```
$ az login
```

4. 一旦浏览器认证完成，您天蓝色的预定信息返回以JSON格式：

```
[
  {
    "cloudName": "AzureCloud",
    "id": "09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx",
    "isDefault": true,
    "name": "Microsoft Azure Enterprise",
    "state": "Enabled",
    "tenantId": "ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx",
    "user": {
      "name": "cisco@cisco.com",
      "type": "user"
    }
  }
]
```

5. 在您开始以配置步骤的其余前，这是一些有用的命令和提示在AzureCLI。

- 对于帮助用可用的子命令和什么他们执行，请使用- h选项。

```
$ az account -h
```

- 默认情况下所有输出返回以JSON格式。对于更加容易的可读性，您能使用--显示的**输出表**选项在表里。

```
$ az account list-locations --output table
```

- 得到所有可用的vm's列表或替换--与过滤表的下面其它选项之一的**All**选项。

```
$ az vm image list --all --output table
```

You are retrieving all the images from server which could take more than a minute. To shorten the wait, provide '--publisher', '--offer' or '--sku'. Partial name search is supported.

- 参考Microsoft的[Azure CLI 2.0](#)文档关于所有配置命令的详细信息。

步骤2.创建资源组。

- 资源组是拿着一个天蓝色的解决方案的相关资源的容器。 给予一个名字您的资源组并且选择位置配置容器。 此示例使用中南部的美国。

```
$ az account list-locations --output table
```

DisplayName	Latitude	Longitude	Name
-----	-----	-----	-----

East Asia	22.267	114.188	eastasia
Southeast Asia	1.283	103.833	southeastasia
Central US	41.5908	-93.6208	centralus
East US	37.3719	-79.8164	eastus
East US 2	36.6681	-78.3889	eastus2
West US	37.783	-122.417	westus
North Central US	41.8819	-87.6278	northcentralus
South Central US	29.4167	-98.5	southcentralus

```
$ az group create --name CorporateDatacenterResourceGroup --location "South Central US"
{
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup",
  "location": "southcentralus",
  "managedBy": null,
  "name": "CorporateDatacenterResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

- 模板(添加资源组)

```
redundancy
cloud provider azure 100
bfd peer
route-table
default-gateway ip
cidr ip
app-key
subscription-id
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

步骤3.创建Vnet。

- Vnet是我们的网络配置IP地址的空间。此范围然后被分裂成更小的子网并且分配到接口。给予一个名字您的vnet，分配它到在创建的资源组第2步并且分配前缀范围。如果不指定前缀，Azure通常分配您10.0.0.0/16。

```
$ az network vnet create --name CorporateDatacenterVnet --resource-group
CorporateDatacenterResourceGroup --address-prefix 192.168.0.0/16
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "192.168.0.0/16"
      ]
    },
    "ddosProtectionPlan": null,
    "dhcpOptions": {
      "dnsServers": []
    },
    "enableDdosProtection": false,
    "enableVmProtection": false,
    "etag": "W/\"7c39a7a9-46e5-4082-a016-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
```

```

xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtual
Networks/CorporateDatacenterVnet",
  "location": "southcentralus",
  "name": "CorporateDatacenterVnet",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceGuid": "3d95d732-e46a-4fae-a34b-xxxxxxxxxxxx",
  "subnets": [],
  "tags": {},
  "type": "Microsoft.Network/virtualNetworks",
  "virtualNetworkPeerings": []
}
}

```

步骤4.创建路由表。

1. 创建内部的对的接口的一个路由表。

```

$ az network route-table create --name InsideRoutetable --resource-group
CorporateDatacenterResourceGroup
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"45088005-cb6f-4356-bb18-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/routeTables"
}

```

模板(添加路由表)

```

redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip
app-key
subscription-id
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup

```

2. 创建外部面对的接口的一个路由表。

```

$ az network route-table create --name OutsideRoutetable --resource-group
CorporateDatacenterResourceGroup
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"a89b6230-9542-468c-b4b2-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/OutsideRoutetable",
  "location": "southcentralus",
  "name": "OutsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],

```

```

    "subnets": null,
    "tags": null,
    "type": "Microsoft.Network/routeTables"
}

```

步骤5.创建子网。

1. 创建从您为在第3步的vnet分配的空间的一个/24子网，然后分配它到内部的路由表。

```

$ az network vnet subnet create --address-prefix 192.168.1.0/24 --name InsideSubnet --
resource-group CorporateDatacenterResourceGroup --vnet-name CorporateDatacenterVnet --
route-table InsideRoutetable
{
  "addressPrefix": "192.168.1.0/24",
  "etag": "W/\"a0dbd178-3a45-48fb-xxxx-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
  "ipConfigurations": null,
  "name": "InsideSubnet",
  "networkSecurityGroup": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceNavigationLinks": null,
  "routeTable": {
    "disableBgpRoutePropagation": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/InsideRoutetable",
    "location": null,
    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}

```

2. 创建从您为vnet分配的空间的另一个/24子网并且分配它到外部路由表。

```

$ az network vnet subnet create --address-prefix 192.168.2.0/24 --name OutsideSubnet --
resource-group CorporateDatacenterResourceGroup --vnet-name CorporateDatacenterVnet --
route-table OutsideRoutetable
{
  "addressPrefix": "192.168.2.0/24",
  "etag": "W/\"874d1019-90a0-44fd-a09c-0aed8f2ede5b\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
  "ipConfigurations": null,
  "name": "OutsideSubnet",
  "networkSecurityGroup": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceNavigationLinks": null,
  "routeTable": {
    "disableBgpRoutePropagation": null,

```



```

    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/OutsideRoutetable",
    "location": null,
    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}

```

步骤6.创建CSR1000v路由器。

每个VM需要有意意味着每个VM 2个NIC的2个接口(内部和外部)。 创建2个NIC并且关联公有IP对外部NIC。

1. 创建公共IP地址。

```

$ az network public-ip create --name CSRAPublicIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\38306703-153b-456b-b2e4-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRA",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "40.124.43.82",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRAPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "610e1631-331a-4971-8502-xxxxxxxxxxxx",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

2. 创建外部NIC并且关联公共IP地址对它。

```

$ az network nic create --name CSRAOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address CSRAPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],

```

```
"dnsServers": [],
"internalDnsNameLabel": null,
"internalDomainNameSuffix": "plk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
"internalFqdn": null
},
"enableAcceleratedNetworking": false,
"enableIpForwarding": false,
"etag": "W/\\"06fd60de-6547-4992-b506-xxxxxxxxxxxx\"",
"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface",
"ipConfigurations": [
  {
    "applicationGatewayBackendAddressPools": null,
    "applicationSecurityGroups": null,
    "etag": "W/\\"06fd60de-6547-4992-xxxx-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface/ipConfigurations/ipconfig1",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.2.4",
    "privateIpAddressVersion": "IPv4",
    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": {
      "dnsSettings": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/CSRAPublicIP",
      "idleTimeoutInMinutes": null,
      "ipAddress": null,
      "ipConfiguration": null,
      "ipTags": null,
      "location": null,
      "name": null,
      "provisioningState": null,
      "publicIpAddressVersion": null,
      "publicIpAllocationMethod": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceGuid": null,
      "sku": null,
      "tags": null,
      "type": null,
      "zones": null
    },
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
```

```

    }
  }
},
"location": "southcentralus",
"macAddress": null,
"name": "CSRAOutsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "93413822-e819-4644-ac0d-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

3. 创建内部的NIC。

```

$ az network nic create --name CSRAInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAInsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.4",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {
          "addressPrefix": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
          "ipConfigurations": null,
          "name": null,
          "networkSecurityGroup": null,
          "provisioningState": null,

```

```

        "resourceGroup": "CorporateDatacenterResourceGroup",
        "resourceNavigationLinks": null,
        "routeTable": null,
        "serviceEndpoints": null
    }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "CSRASideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "0f7ae52a-47c3-4563-9fe0-b1484e88296e",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

4. 列出在Azure的可用的CSR1000v镜像。此示例使用cisco:cisco-csr-1000v:16_7:16.7.120171201的缸名字。

```

az vm image list --all --publisher Cisco --offer cisco-csr-1000v
[
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_5",
    "urn": "cisco:cisco-csr-1000v:16_5:16.5.120170418",
    "version": "16.5.120170418"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_5",
    "urn": "cisco:cisco-csr-1000v:16_5:16.5.220171128",
    "version": "16.5.220171128"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_6",
    "urn": "cisco:cisco-csr-1000v:16_6:16.6.120170804",
    "version": "16.6.120170804"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_6",
    "urn": "cisco:cisco-csr-1000v:16_6:16.6.220171219",
    "version": "16.6.220171219"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_7",
    "urn": "cisco:cisco-csr-1000v:16_7:16.7.120171201",
    "version": "16.7.120171201"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "3_16",

```

```

    "urn": "cisco:cisco-csr-1000v:3_16:3.16.420170208",
    "version": "3.16.420170208"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "3_16",
    "urn": "cisco:cisco-csr-1000v:3_16:3.16.520170215",
    "version": "3.16.520170215"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "csr-azure-byol",
    "urn": "cisco:cisco-csr-1000v:csr-azure-byol:16.40.120170206",
    "version": "16.40.120170206"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "csr-azure-byol",
    "urn": "cisco:cisco-csr-1000v:csr-azure-byol:3.16.0",
    "version": "3.16.0"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "csr-azure-byol",
    "urn": "cisco:cisco-csr-1000v:csr-azure-byol:3.16.2",
    "version": "3.16.2"
  }
]

```

5. 配置与镜像的缸名字的CSR1000v。

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRA --location
southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics
CSRAOutsideInterface CSRAInsideInterface --admin-username cisco --admin-password
"Cisco1234567" --authentication-type password
Running ..
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/vi
rtualMachines/CSRA",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-83-58,00-0D-3A-5D-89-27",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.4,192.168.1.4",
  "publicIpAddress": "40.124.43.82",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

在几分钟之后，新的CSR1000v启动。

```

$ az vm list --resource-group CorporateDatacenterResourceGroup --show-details --output
table

```

Name	ResourceGroup	PowerState	PublicIps	Fqdns	Location	Zones
CSRA	CorporateDatacenterResourceGroup	VM running	40.124.43.82		southcentralus	

6. 登陆对CSR1000v并且验证功能。

```
$ ssh cisco@40.124.43.82
The authenticity of host '40.124.43.82 (40.124.43.82)' can't be established.
RSA key fingerprint is SHA256:q33FHw7RlkDn
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.124.43.82' (RSA) to the list of known hosts.
Password:
```

```
CSRA#
CSRA#show ip interface brief
Interface IP-Address OK? Method Status Protocol
GigabitEthernet1 192.168.2.4 YES DHCP up up
GigabitEthernet2 192.168.1.4 YES DHCP up up
```

步骤7.创建第二个CSR1000v路由器。

1. 创建公共IP地址。

```
$ az network public-ip create --name CSRBPUBLICIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\ "f0f98dac-ea56-4efe-8da6-81a221ac3474\\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRBPUBLICIP",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "23.100.122.102",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRBPUBLICIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "aa03bc26-22df-4696-bd77-ca29df029d7d",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}
```

2. 创建外部NIC并且关联公共IP地址对它。

```
$ az network nic create --name CSRBOURSIDEINTERFACE --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVNET --
public-ip-address CSRBPUBLICIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
```

```
"enableIpForwarding": false,
"etag": "W/\\"ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBOoutsideInterface",
"ipConfigurations": [
  {
    "applicationGatewayBackendAddressPools": null,
    "applicationSecurityGroups": null,
    "etag": "W/\\"ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBOoutsideInterface/ipConfigurations/ipconfig1",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.2.5",
    "privateIpAddressVersion": "IPv4",
    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": {
      "dnsSettings": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/CSRBPUBLICIP",
      "idleTimeoutInMinutes": null,
      "ipAddress": null,
      "ipConfiguration": null,
      "ipTags": null,
      "location": null,
      "name": null,
      "provisioningState": null,
      "publicIpAddressVersion": null,
      "publicIpAllocationMethod": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceGuid": null,
      "sku": null,
      "tags": null,
      "type": null,
      "zones": null
    },
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
},
"location": "southcentralus",
"macAddress": null,
"name": "CSRBOoutsideInterface",
```

```

    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "c3f05156-ad07-4abd-a006-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

3. 创建内部的NIC。

```

$ az network nic create --name CSRBIInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i01lcksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"15edf738-fc77-431c-80f3-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBIInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"15edf738-fc77-431c-80f3-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBIInsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.5",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {
          "addressPrefix": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
          "ipConfigurations": null,
          "name": null,
          "networkSecurityGroup": null,
          "provisioningState": null,
          "resourceGroup": "CorporateDatacenterResourceGroup",
          "resourceNavigationLinks": null,
          "routeTable": null,
          "serviceEndpoints": null
        }
      }
    ]
  },
}

```



```

    "location": "southcentralus",
    "macAddress": null,
    "name": "CSRBIInsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "085c88fc-9e78-49be-a5a7-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

4. 配置与同一个镜像cisco:cisco-csr-1000v:16_7:16.7.120171201的第二个CSR1000v。

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRBI --location southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics CSRBIOutsideInterface CSRBIInsideInterface --admin-username cisco --admin-password "Cisco1234567" --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/virtualMachines/CSRBI",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-8C-51,00-0D-3A-5D-85-2A",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.5,192.168.1.5",
  "publicIpAddress": "23.100.122.102",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

步骤8.用在第6.步的同一个程序创建主机VM。此示例使用UbuntuLTS。

1. 创建公共IP地址。

```

$ az network public-ip create --name VMHostPublicIP --resource-group CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"5943a230-1eeb-4cf0-b856-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/VMHostPublicIP",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "104.215.77.207",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "VMHostPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "ea19c10a-2fd3-498f-b984-xxxxxxxxxxxx",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

```
}  
}
```

2. 创建外部NIC并且关联OutsideSubnet和公共IP地址对它。当子网与NIC时产生关联，IP地址自动地分配到NIC。在本例中，OutsideSubnet是192.168.2.0/24，并且IP地址自动地分配到NIC是192.168.2.6。

```
$ az network nic create --name VMHostOutsideInterface --resource-group  
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --  
public-ip-address VMHostPublicIP  
{  
  "NewNIC": {  
    "dnsSettings": {  
      "appliedDnsServers": [],  
      "dnsServers": [],  
      "internalDnsNameLabel": null,  
      "internalDomainNameSuffix": "gzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",  
      "internalFqdn": null  
    },  
    "enableAcceleratedNetworking": false,  
    "enableIpForwarding": false,  
    "etag": "W/\\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",  
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-  
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne  
tworkInterfaces/VMHostOutsideInterface",  
    "ipConfigurations": [  
      {  
        "applicationGatewayBackendAddressPools": null,  
        "applicationSecurityGroups": null,  
        "etag": "W/\\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",  
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-  
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne  
tworkInterfaces/VMHostOutsideInterface/ipConfigurations/ipconfig1",  
        "loadBalancerBackendAddressPools": null,  
        "loadBalancerInboundNatRules": null,  
        "name": "ipconfig1",  
        "primary": true,  
        "privateIpAddress": "192.168.2.6",  
        "privateIpAddressVersion": "IPv4",  
        "privateIpAllocationMethod": "Dynamic",  
        "provisioningState": "Succeeded",  
        "publicIpAddress": {  
          "dnsSettings": null,  
          "etag": null,  
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-  
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu  
blicIPAddresses/VMHostPublicIP",  
          "idleTimeoutInMinutes": null,  
          "ipAddress": null,  
          "ipConfiguration": null,  
          "ipTags": null,  
          "location": null,  
          "name": null,  
          "provisioningState": null,  
          "publicIpAddressVersion": null,  
          "publicIpAllocationMethod": null,  
          "resourceGroup": "CorporateDatacenterResourceGroup",  
          "resourceGuid": null,  
          "sku": null,  
          "tags": null,  
          "type": null,  
          "zones": null  
        },  
        "resourceGroup": "CorporateDatacenterResourceGroup",  
        "subnet": {
```

```

    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "VMHostOutsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "89588a04-6ba6-467d-a86f-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

3. 创建内部的NIC。

```

$ az network nic create --name VMHostInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\ "dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\" ",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\ "dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\" ",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostInsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.6",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",

```

```

    "provisioningState": "Succeeded",
    "publicIpAddress": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "subnet": {
      "addressPrefix": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
      "ipConfigurations": null,
      "name": null,
      "networkSecurityGroup": null,
      "provisioningState": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null,
      "serviceEndpoints": null
    }
  },
  "location": "southcentralus",
  "macAddress": null,
  "name": "VMHostInsideInterface",
  "networkSecurityGroup": null,
  "primary": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceGuid": "8ef12cdd-cc31-432e-99cf-xxxxxxxxxxxx",
  "tags": null,
  "type": "Microsoft.Network/networkInterfaces",
  "virtualMachine": null
}
}

```

4. 配置Ubuntu VM。此示例使用UbuntuLTS。

```
az vm image list --output table
```

You are viewing an offline list of images, use --all to retrieve an up-to-date list

Offer	Publisher	Skus	Urn
UrnAlias	Version		
CentOS	OpenLogic	7.3	OpenLogic:CentOS:7.3:latest
CentOS	latest		
CoreOS	CoreOS	Stable	CoreOS:CoreOS:Stable:latest
CoreOS	latest		
Debian	credativ	8	credativ:Debian:8:latest
Debian	latest		
openSUSE-Leap	SUSE	42.3	SUSE:openSUSE-Leap:42.3:latest
openSUSE-Leap	latest		
RHEL	RedHat	7.3	RedHat:RHEL:7.3:latest
RHEL	latest		
SLES	SUSE	12-SP2	SUSE:SLES:12-SP2:latest
SLES	latest		
UbuntuServer	Canonical	16.04-LTS	Canonical:UbuntuServer:16.04-
LTS:latest		UbuntuLTS	latest
WindowsServer	MicrosoftWindowsServer	2016-Datacenter	Win2016Datacenter latest
MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest			
WindowsServer	MicrosoftWindowsServer	2012-R2-Datacenter	Win2012R2Datacenter latest
MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest			
WindowsServer	MicrosoftWindowsServer	2012-Datacenter	Win2012Datacenter latest
MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest			
WindowsServer	MicrosoftWindowsServer	2008-R2-SP1	Win2008R2SP1 latest
MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest			

```
$ az vm create --resource-group CorporateDatacenterResourceGroup --name VmHost --location southcentralus --image UbuntuLTS --admin-user cisco --admin-password Cisco1234567 --nics VMHostOutsideInterface VMHostInsideInterface --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/virtualMachines/VmHost",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-B7-CB,00-0D-3A-5D-B8-9B",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.6,192.168.1.6",
  "publicIpAddress": "104.215.77.207",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}
```

步骤9.添加路由到路由表和VMs。

1. 添加内部的子网的默认路由到路由流量通过CSR A通过设置下一跳IP地址作为192.168.1.4。这在InsideRouteTable执行。

```
$ az network route-table route create --address-prefix 8.8.8.8/32 --name default_route --next-hop-type VirtualAppliance --resource-group CorporateDatacenterResourceGroup --route-table-name InsideRouteTable --next-hop-ip-address 192.168.1.4
{
  "addressPrefix": "8.8.8.8/32",
  "etag": "W/\\"ef9e650a-5d70-455d-b958-5a0efc07e7ad\\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRouteTable/routes/default_route",
  "name": "default_route",
  "nextHopIpAddress": "192.168.1.4",
  "nextHopType": "VirtualAppliance",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup"
}
```

2. 添加数据流的路由在网络到达OutsideRouteTable的互联网。

```
$ az network route-table route create --address-prefix 8.8.8.8/32 --name internet --next-hop-type Internet --resource-group CorporateDatacenterResourceGroup --route-table-name OutsideRouteTable
{
  "addressPrefix": "8.8.8.8/32",
  "etag": "W/\\"d2c7e32e-8d32-4856-a3a6-xxxxxxxxxxxx\\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/OutsideRouteTable/routes/internet",
  "name": "internet",
  "nextHopIpAddress": null,
  "nextHopType": "Internet",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup"
}
```

3. 登陆对Ubuntu VM并且添加路由通过内部接口强制数据流到8.8.8.8。天蓝色的路由表在子网自动地使用第一个IP作为其网关。意味着的内部接口的(eth1)子网是192.168.1.0/24 192.168.1.1是主机VM的默认千兆瓦地址。

```
$ ifconfig
```

```
eth0 Link encap:Ethernet HWaddr 00:0d:3a:5d:b7:cb
inet addr:192.168.2.6 Bcast:192.168.2.255 Mask:255.255.255.0
inet6 addr: fe80::20d:3aff:fe5d:b7cb/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3986 errors:0 dropped:0 overruns:0 frame:0
TX packets:2881 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3475393 (3.4 MB) TX bytes:592740 (592.7 KB)
```

```
eth1 Link encap:Ethernet HWaddr 00:0d:3a:5d:b8:9b
inet addr:192.168.1.6 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::20d:3aff:fe5d:b89b/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2 errors:0 dropped:0 overruns:0 frame:0
TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:762 (762.0 B) TX bytes:1620 (1.6 KB)
```

```
$ sudo route add -host 8.8.8.8 gw 192.168.1.1 dev eth1
$ route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.2.1	0.0.0.0	UG	0	0	0	eth0
8.8.8.8	192.168.1.1	255.255.255.255	UGH	0	0	0	eth1
168.63.129.16	192.168.2.1	255.255.255.255	UGH	0	0	0	eth0
169.254.169.254	192.168.2.1	255.255.255.255	UGH	0	0	0	eth0
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

模板(添加cidr ip)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key
subscription-id
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

Note:在的第10步CSR1000v路由器必须配置NAT为了连接互联网(8.8.8.8)。 **Note:**步骤10-14包括CSR1000v路由器的配置HA的。从[Cisco CSR 1000v部署指南](#)的缩写的步骤[Microsoft Azure](#)是提供的初期从配置Trustpool。访问指南关于完全详细资料。

步骤10.配置CSR1000v路由器。

1. 配置在两CSR1000v路由器的Trustpool

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#crypto pki trustpool import url
http://www.cisco.com/security/pki/trs/ios.p7b
Reading file from http://www.cisco.com/security/pki/trs/ios.p7b
Loading http://www.cisco.com/security/pki/trs/ios.p7b !!!
% PEM files import succeeded.
```

2. 配置在Cisco CSR 1000v路由器和enable (event)双向转发检测(BFD)和路由协议之间的

IPSec隧道(EIGRP或BGP)在路由器之间的隧道对等体故障检测的。 **Note:**在配置的隧道目的地地址是对等体CSR的公共IP地址。 CSR配置

```
crypto isakmp policy 1
  encr aes 256
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0
!
crypto ipsec transform-set uni-perf esp-aes 256 esp-sha-hmac
  mode tunnel
!
crypto ipsec profile vti-1
  set security-association lifetime kilobytes disable
  set security-association lifetime seconds 86400
  set transform-set uni-perf
  set pfs group2
!
interface Tunnell
  ip address 192.168.101.1 255.255.255.252
  bfd interval 500 min_rx 500 multiplier 3
  tunnel source GigabitEthernet1
  tunnel mode ipsec ipv4
  tunnel destination 23.100.122.102 /* Public IP of the peer CSR */
  tunnel protection ipsec profile vti-1
!
router eigrp 1
bfd all-interfaces
network 192.168.101.0
```

CSRB配置

```
crypto isakmp policy 1
  encr aes 256
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0
!
crypto ipsec transform-set uni-perf esp-aes 256 esp-sha-hmac
  mode tunnel
!
crypto ipsec profile vti-1
  set security-association lifetime kilobytes disable
  set security-association lifetime seconds 86400
  set transform-set uni-perf
  set pfs group2
!
interface Tunnell
  ip address 192.168.101.2 255.255.255.252
  bfd interval 500 min_rx 500 multiplier 3
  tunnel source GigabitEthernet1
  tunnel mode ipsec ipv4
  tunnel destination 40.124.43.82 /* Public IP of the peer CSR */
  tunnel protection ipsec profile vti-1
!
router eigrp 1
bfd all-interfaces
network 192.168.101.0
```

3. NAT的相同配置和路由在两CSR1000v路由器使用。这是为VM互联网可到达性通过内部接口

```
o
interface GigabitEthernet1
  ip nat outside
!
interface GigabitEthernet2
  ip nat inside
!
ip nat inside source list 10 interface GigabitEthernet1 overload
```

```

access-list 10 permit 192.168.1.0 0.0.0.255 /* Translating the inside subnet of the VM */
!
ip route 0.0.0.0 0.0.0.0 192.168.2.1
ip route 192.168.1.0 255.255.255.0 GigabitEthernet2 192.168.1.1

```

4. 添加访问控制(IAM)路由表的。在AzureCLI，请允许应用程序(CSRA和CSRB)在故障切换期间，修改在Azure的InsideRouteTable。注释作为将使用的InsideRouteTable的id--在下一个部分的范围选项。

```

$ az network route-table show --resource-group CorporateDatacenterResourceGroup --name
InsideRoutetable
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\f0c85464-bba0-465a-992a-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
  ...

```

模板(添加订阅id)

```

redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup

```

5. 创建InsideRouteTable的IAM角色。--范围选项从从早先输出的ID字段被采取。注释是APP KEY)的APP id、密码(和承租人id)。

```

$ az ad sp create-for-rbac -n "InsideRouteTableIAM" --role "network contributor" --scopes
/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRoutetable --years 2099
{
  "appId": "576dd4f1-c08d-xxxx-xxxx-xxxxxxxxxxxx",
  "displayName": "InsideRouteTableIAM",
  "name": "http://InsideRouteTableIAM",
  "password": "aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxx",
  "tenant": "ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx"
}

```

模板(添加APP KEY、APP id和承租人id)

```

redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup

```

6. 配置在两路由器的网云冗余。配置之间的唯一的区别在两路由器是bfd对等体和默认网关。

CSRA配置

```

redundancy
cloud provider azure 100
bfd peer 192.168.101.2

```



```

route-table InsideRoutetable
default-gateway ip 192.168.1.4
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup

```

CSRB配置

```

redundancy
cloud provider azure 100
bfd peer 192.168.101.1
route-table InsideRoutetable
default-gateway ip 192.168.1.5
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup

```

验证高可用性

1. 检查BFD和网云配置。

```

CSRA#show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	192.168.2.4	YES	DHCP	up	up
GigabitEthernet2	192.168.1.4	YES	DHCP	up	up
Tunnell	192.168.101.1	YES	manual	up	up

```

CSRB#show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	192.168.2.5	YES	DHCP	up	up
GigabitEthernet2	192.168.1.5	YES	DHCP	up	up
Tunnell	192.168.101.2	YES	NVRAM	up	up

```

CSRA#show bfd neighbors

```

```

IPv4 Sessions

```

NeighAddr	LD/RD	RH/RS	State	Int
192.168.101.2	4097/4097	Up	Up	Tu1

```

CSRA#show redundancy cloud provider azure 100
Cloud HA: work_in_progress=FALSE
Provider : AZURE node 100
State : idle
BFD peer      = 192.168.101.2
BFD intf      = Tunnell
resource group = CorporateDatacenterResourceGroup
subscription id = 09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxxxx
tenant id = ae49849c-2622-4d45-b95e-xxxxxxxxxxxxxx
application id = 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxxxxxxx
application key = aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxxx
route-table    = InsideRoutetable
cidr           = 8.8.8.8/32
Default Gateway IP = 192.168.1.4

```

2. 从VM运行ping和traceroute到目的地。保证ping是通过内部的eth1接口。

```

$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.6 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=54 time=10.5 ms

```

```
64 bytes from 8.8.8.8: icmp_seq=2 ttl=54 time=10.6 ms
```

```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 192.168.1.4 (192.168.1.4) 1.516 ms 1.503 ms 1.479 ms
```

```
cisco@VmHost:~$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.6 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=10.2 ms
```

3. Traceroute表示，从VM的路径到8.8.8.8是通过CSRA内部接口。

```
cisco@VmHost:~$ sudo traceroute -I 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 192.168.1.4 (192.168.1.4) 34.003 ms 34.000 ms 33.998 ms
```

4. 关闭CSRA的隧道1接口模拟故障切换。

```
CSRA#config t
Enter configuration commands, one per line. End with CNTL/Z.
CSRA(config)#int tunnel1
CSRA(config-if)#sh
```

5. 注意到数据流当前流经CSRB的专用接口。

```
cisco@VmHost:~$ sudo traceroute -I 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 192.168.1.5 (192.168.1.5) 1.294 ms 1.291 ms 1.290 ms
```

Note:当发生故障时，天蓝色的网云可能引入延迟。延迟不再比1分钟应该是。

Troubleshoot

- 观察消息的关闭调试在HA故障切换期间。

```
CSRA#debug redundancy cloud all
CSRA#debug ip http all
```

- 允许CSR1000v做API呼叫到天蓝色的路由表的认证和凭证错误归结于无效访问控制。仔细检查适当的id's在第10.步被配置。

```
*Jul 13 23:29:53.365: CLOUD-HA : res content iov_len=449
iov_base={"error":"invalid_client","error_description":"AADSTS70002:
Error validating credentials. AADSTS50012: Invalid client secret is provided.\r\nTrace ID:
56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx\r\n
Correlation ID: cce94817-29eb-4ebd-833a-\r\nTimestamp: 2018-07-13
23:29:54Z","error_codes":[70002,50012],"timestamp":"2018-07-13
23:29:54Z","trace_id":"56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx","correlation_id":"cce94817-29eb-
4ebd-833a"}
```

Related Information

- [Azure CLI 2.0](#)
- [Cisco CSR 1000v Microsoft Azure部署指南](#)
- [选择Azure和并行的Azure CLI和PowerShell命令的正确的输出的装饰](#)